Adjoint Computational Electromagnetics

Mohamed Kamal Abd Elrahman
6 October, Giza,
Egypt

October 22, 2019

The adjoint method is used to inverse design linear and nonlinear photonic devices. The purpose of this handout is give the reader an introduction to the topic.

Motivation

In normal electromagnetic simulation, we are given a material distribution represented by $(\epsilon(x,y,z), \mu(x,y,z))$ and a current density source $\mathbf{J}(\mathbf{x},\mathbf{y},\mathbf{z})$. Maxwell's equations are then solved using one of the computational techniques (FD, FEM, MoM) for the excited electromagnetic field $(\mathbf{H},\mathbf{E})$. For linear, isotropic, non magnetic and stationary media [1] $\mathbf{B} = \mu_0\mathbf{H}$ and $\mathbf{D} = \epsilon\mathbf{E}$. Maxwell's equations take the form

$$\nabla \times \mathbf{E} = -\mu_0 \frac{\partial \mathbf{H}}{\partial t} \tag{1a}$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \epsilon \frac{\partial \mathbf{E}}{\partial t} \tag{1b}$$

[1] In general ϵ and μ may be nonlinear anisotropic tensors. We restricted ourselves to linear isotropic media for simplicity and most material are nearly the case.

We are asked to maximize the scaler quantity $G(\mathbf{E},\mathbf{H})$ by varying the dielectric structure $\epsilon(x,y,z)$. G is often a measure of device performance: for example, the power flowing in certain direction or mode conversion efficiency. This is an inverse design problem, we start by defining an objective G and ask for the best material distribution to achieve this objective.

The gradient quantity $\frac{dG}{d\epsilon}$ is useful to compute. It could be used as a measure of robustness and sensitivity to any structure deformations like fabrication imperfections. In inverse design, it is the major quantity needed as it points in the direction of steepest ascent, from which we have an informed decision of how to update the dielectric structure.

$$\epsilon \longleftarrow \epsilon + \alpha \frac{dG}{d\epsilon} \tag{2}$$

Where α is the step size in the gradient direction[2].

[2] Some algorithms attempt to optimize the step size so that the step maximally increases G

Assuming we have n number of design points, to calculate the sensitivity $\frac{dG}{d\epsilon}$ there are different approaches which vary in complexity and accuracy. One way is by using a finite difference approximation $\frac{dG}{d\epsilon} \approx \frac{\Delta G}{\Delta \epsilon}$. In this approach we need to carry n number of forward simulations. For a typical $100 \times 100 \times 100$ 3D finite difference grid,

this means 1000000 simulations!. This approach also suffers from numerical discretization errors and the need to determine how small or large is $\Delta\epsilon$. Automatic differentiation can be used to get the derivative exactly. However, there are are two modes in automatic differentiation. The automatic forward mode, like finite difference, needs n number of forward simulations. In the automatic adjoint or reverse mode, two simulations are only needed regardless of the number of the design points. So, the adjoint method is by far much better in accuracy than finite difference and computational cost than automatic forward differentiation.

Problem Formulation

Before driving the adjoint equation, lets put Maxwell's in a matrix form:

$$\begin{bmatrix} \mu_0 & 0 \\ 0 & -\epsilon \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{H}}{\partial t} \\ \frac{\partial \mathbf{E}}{\partial t} \end{bmatrix} + \begin{bmatrix} 0 & \nabla\times \\ \nabla\times & 0 \end{bmatrix} \begin{bmatrix} \mathbf{H} \\ \mathbf{E} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{J} \end{bmatrix} \tag{3}$$

Which could be written as

$$C\frac{d\mathbf{X}}{dt} + D\mathbf{X} = \mathbf{S} \tag{4}$$

if the following definitions have been made $\mathbf{X} = \begin{bmatrix} \mathbf{H} \\ \mathbf{E} \end{bmatrix}$, $C = \begin{bmatrix} \mu_0 & 0 \\ 0 & -\epsilon \end{bmatrix}$, $D = \begin{bmatrix} 0 & \nabla\times \\ \nabla\times & 0 \end{bmatrix}$ and $\mathbf{S} = \begin{bmatrix} 0 \\ \mathbf{J} \end{bmatrix}$

The initial condition for equation(4) is

$$I(\mathbf{X}(0), \epsilon) = \mathbf{0} \tag{5}$$

Our optimization problem can be then formulated as

$$\begin{aligned} \max_{\epsilon} \quad & G(\epsilon) = \int_0^T g(\mathbf{X}, \epsilon, t)dt \\ \text{s.t.} \quad & C\frac{d\mathbf{X}}{dt} + D\mathbf{X} - \mathbf{S} = 0 \\ & I(\mathbf{X}(0), \epsilon) = \mathbf{0} \end{aligned} \tag{6}$$

Where we have assumed our objective function can be written in this integral form $\int_0^T g(\mathbf{X}, \epsilon, t)dt$.

Equation (4) is commonly solved by discretization in space first and then solving the resulting system of Ordinary Differential Equations (ODEs). We can either start by replacing time derivatives with finite differences to arrive at a recursive set of spatial problems to be discretized. The discretization in space could be in done in real or spectral domains. Maxwell's equation could even be expanded in the natural complete eigen-basis of the problem in a coupled-mode-like

fashion and inverse design could be done with very low dimensional matrices. Among the numerical methods, the finite difference method, the finite element method, and the method of moments form the core of computational electromagnetics. One of our goals is to show a simple one dimensional example with code for each method along with its adjoint formulation. Interested readers can directly add an adjoint solver on top their EM solvers very easily and start doing inverse design.

Adjoint Method

In gradient based inverse design, we start with an initial dielectric profile $\epsilon^0(x, y, z)$ and then use local gradient of the objective function to guide the search in device space [3]. The gradient of the objective function G in equation (6)

$$\frac{dG}{d\epsilon} = \int_0^T \left[\frac{\partial g}{\partial \epsilon} + \frac{\partial g}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \epsilon} \right] dt \tag{7}$$

[3] Additional constraints should be added to guarantee robustness and easy fabrication of the final discovered devices. Fabrication constraints play a key role in inverse design and we will talk about it more in a future separate handout.

To evaluate the last integral, we need to calculate the term $\frac{\partial \mathbf{X}}{\partial \epsilon}$. If the vector ϵ is n dimensional, we need to repeat the simulation n number of times. For each, we change the component ϵ_i by $\Delta\epsilon_i$, then the change in the objective function is determined. Finally, the slope is calculated by dividing the change in the objective function by $\Delta\epsilon_i$. The whole point of adjoint method is to avoid calculating the derivative term $\frac{\partial \mathbf{X}}{\partial \epsilon}$. Soon, we will discover that the adjoint method can magically determine the gradient at all design points with an additional cost of one extra simulation. The remaining terms inside the objective integral are not really problematic. $\frac{\partial g}{\partial \mathbf{X}}$ and $\frac{\partial g}{\partial \mathbf{X}}$ can be evaluated (efficiently and exactly) with a reverse mode automatic differentiation package [4]. In Julia, the package "Nabla.jl" does a pretty good job in calculating these gradients.

[4] There are tow levels of reverse mode differentiation (1) Differentiation of scaler function with respect to large dimensional vectors. (2) Efficient evaluation of PDE constrained objective function gradients using adjoint method.

To eliminate the term $\frac{\partial \mathbf{X}}{\partial \epsilon}$, the first step is to add multiples of 0 to the objective function.

$$G = \int_0^T g(\mathbf{X}, \epsilon, t) + \lambda^T \left(C\frac{d\mathbf{X}}{dt} + D\mathbf{X} - \mathbf{S}) \right) + \kappa^T \left(I(\mathbf{X}(0), \epsilon) \right) dt \tag{8}$$

The vector λ is time dependent. κ is another vector associated with the initial conditions. The objective function has not changed after this modification, the equation is also the same for any choice of λ and κ. We will choose them in a way that eliminate the need for the term $\frac{\partial \mathbf{X}}{\partial \epsilon}$ in the objective function gradient $\frac{dG}{d\epsilon}$. The vectors λ and κ are commonly called Lagrange multipliers and the modified objective function is the system Lagrangian that we seek to maximize.

Taking the total derivative of the Lagrangian,

$$\frac{dG}{d\epsilon} = \int_0^T \left(\left[\frac{\partial g}{\partial \epsilon} + \frac{\partial g}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \epsilon} \right] + \lambda^T \left[\frac{dC}{d\epsilon} \frac{d\mathbf{X}}{dt} + C \frac{\partial}{\partial \epsilon} \frac{d\mathbf{X}}{dt} + D \frac{\partial \mathbf{X}}{\partial \epsilon} \right] \right) dt + \kappa^T \left[\frac{\partial I}{\partial \mathbf{X}(0)} \frac{\partial \mathbf{X}(0)}{\partial \epsilon} + \frac{\partial I}{\partial \epsilon} \right] \quad (9)$$

The problematic term $\frac{\partial \mathbf{X}}{\partial \epsilon}$ is what we want to git rid off, however we have now its time derivative added. Integration by parts can help us reduce the time derivative to zero order.

$$\int_0^T \left(\lambda^T C \frac{\partial}{\partial \epsilon} \frac{d\mathbf{X}}{dt} \right) dt = \lambda^T C \frac{\partial \mathbf{X}}{\partial \epsilon} |_0^T - \int_0^T \frac{d\left(\lambda^T C\right)}{dt} \frac{\partial \mathbf{X}}{\partial \epsilon} dt \quad (10)$$

Substituting (10) into (9) and collecting terms in $\frac{\partial \mathbf{X}}{\partial \epsilon}$ and $\frac{\partial \mathbf{X}(0)}{\partial \epsilon}$ yield

$$\frac{dG}{d\epsilon} = \int_0^T \left(\left[\frac{\partial g}{\partial \epsilon} + \lambda^T \frac{dC}{d\epsilon} \frac{d\mathbf{X}}{dt} \right] + \left[\frac{\partial g}{\partial \mathbf{X}} + \lambda^T D - \frac{d\left(\lambda^T C\right)}{dt} \right] \frac{\partial \mathbf{X}}{\partial \epsilon} \right) dt + \left[\kappa^T \frac{\partial I}{\partial \mathbf{X}(0)} - \lambda^T C \right] \frac{\partial \mathbf{X}(0)}{\partial \epsilon} + \lambda^T C \frac{\partial \mathbf{X}(T)}{\partial \epsilon} + \kappa^T \frac{\partial I}{\partial \epsilon} \quad (11)$$

We can get rid of $\frac{\partial \mathbf{X}}{\partial \epsilon}$ by choosing λ^T to satisfy the following differential equation

$$\frac{\partial g}{\partial \mathbf{X}} + \lambda^T D - \frac{d\left(\lambda^T C\right)}{dt} = 0 \quad (12)$$

We can get rid of $\frac{\partial \mathbf{X}(0)}{d\epsilon}$ by choosing $\kappa^T = \lambda^T C \left[\frac{\partial I}{\partial \mathbf{X}(0)} \right]^{-1}$

If we also imposed the terminal condition $\lambda^T(T)C = 0$ and noting that in most cases, the initial conditions don't depend on the the material distribution $\frac{\partial I}{\partial \epsilon} = 0$. The gradient of objective function is then

$$\frac{dG}{d\epsilon} = \int_0^T \left[\frac{\partial g}{\partial \epsilon} + \lambda^T \frac{dC}{d\epsilon} \frac{d\mathbf{X}}{dt} \right] dt \quad (13)$$

The algorithm for computing $\frac{dG}{d\epsilon}$ follows:

1. Solve the forward problem for $\mathbf{X}$ from 0 to T

$$C \frac{d\mathbf{X}}{dt} + D\mathbf{X} - \mathbf{S} = 0$$
$$I(\mathbf{X}(0), \epsilon) = \mathbf{0}$$

2. Solve the adjoint problem for λ from T to 0

$$C^T \frac{d\lambda}{dt} - D^T \lambda = \left[\frac{\partial g}{\partial \mathbf{X}} \right]^T$$
$$\lambda(T) = 0$$

3. Get the objective function gradient

$$\frac{dG}{d\epsilon} = \int_0^T \left[\frac{\partial g}{\partial \epsilon} + \lambda^T \frac{dC}{d\epsilon} \frac{d\mathbf{X}}{dt} \right] dt \tag{14}$$

It worth having these points in mind:

- The problem of calculating the gradient with respect to all design variables has been reduced to solving one extra simulation backward in time.
- The adjoint equation is very similar to the forward equation, except the operators have been replaced by their adjoint (which is the same as t transpose for real operators). Thats why the problem is called the adjoint problem. For reciprocal systems, the matrices re equal to their adjoint and EM solvers already developed could be readily extended to solve the adjoint problem.
- The adjoint problem has a different source term, which depends on the gradient of the forward problem solution. This would require storing the field $\mathbf{X}$ at all times. This is a big challenge for numerical methods like FDTD, which usually take long running simulation times.
- The adjoint equation could be converted to an initial value problem by substituting t with $T - \tau$.
- The objective integral needs the derivative of the forward field, which also requires the storage the storage at all times.